

Dynamic Queue Behavior in Networks with Window Protocols

John G. Waclawsky
Department of Computer Science
University of Maryland
College Park MD 20742
and IBM National Service Division
Gaithersburg MD 20879
(301)-240-8378

Ashok K. Agrawala
Department of Computer Science
University of Maryland
College Park MD 20742
(301)-454-4968
agrawala@brillig.umd.edu

In this paper we employ a deterministic analysis technique to characterize the dynamic queueing aspects of window protocols. The deterministic behavior of these protocols and the deterministic influence of the resources along the physical path are explicitly considered in the evaluation of path queue behavior. Transient and steady state queue behavior of fixed and sliding window protocols are investigated. We discover the existence of significant nonlinearities in the dynamics of queue activity.

Window protocols are viewed as logical simplex pipes. These pipes connect a sender and a receiver through a series of heterogeneous physical resources which provide a path of finite delay between them. Links and nodes make up the path resources which supply physical connectivity. The resource with the largest delay is called the bottleneck resource.

Dynamic queue behavior is obtained by explicitly considering the fact that feedback mechanisms employed by window protocols make them inherently cyclic. Thus a group of packets, called a window, enters the network every cycle. The concept of a win-

dow can be formalized in terms of containers which are made available to carry packets through a path. Packets cannot be transmitted without a container. Controlling the number of containers available at the protocol sender controls the amount of data flowing in the path.

Packet transmission by the sender, using the first link, can occur when either the link changes its status from busy to free or an acknowledgement is received. The sender is considered "greedy" since fundamental sender operation is to transmit as long as both packets and containers are available.

Deterministic behavior occurs whenever the arrival rate of packets to the sender is such that there is always a packet available for transmission. This situation occurs frequently in networks for all types of traffic. In fact, the whole class of "Batch" traffic satisfies this arrival situation because of the rapid generation of packets by batch applications. The following assumptions were employed for this analysis.

1. The path is initially empty.
2. Packets are always available for transmission by the sender. Thus data flow only stops when the sender expires his container supply.
3. All packets (including those containing a request or acknowledgement) are the same size.

4. No cross traffic is present.
5. There is no loss or reordering of packets.
6. All resources follow a work conserving discipline.
7. We define that departures from one resource occur at the same time instant as arrivals to the next resource.

Fundamental packet and resource activity shows that the bottleneck exerts a major influence on path behavior. This is seen for two reasons. First, when load is heavy, packets depart from the path under control of the bottleneck. Thus, the bottleneck controls path throughput. Second, if a packet is delayed anywhere along the path it also waits at the bottleneck. Thus, the bottleneck controls the timing of window protocol acknowledgements and all resource utilizations.

The queue formation process is seen as a by-product of the heterogeneous delays that exist along a path. Whenever a higher speed resource exists at the sender, then queue sizes increase normally at slower resources along a path during any period of continuous sender transmission. Clearly, if path resource delays are equal along a path or a slower resource exists "upstream", then no queue buildup can occur "downstream" from the slower or equal speed resource. Thus, queue build up along a path only occurs at, or prior to, the bottleneck location. Once the path is full, whenever both the bottleneck and the protocol sender are transmitting, then packet build up along the path occurs at the same rate that containers are consumed at the sender.

Since the arrival rate of packets to any queue is limited by the slowest upstream resource in the path, we only examine paths with increasing resource delays. Paths without these *exact* characteristics do make up a substantial portion of many actual network environments. Queues within these paths can be analyzed by looking further upstream for an appropriate arrival rate. This is done by shifting packet arrival times through the use of a constant for each queue.

Results show that window protocol activity, along with physical path delays and the value of the window size, controls both the magnitude of queue sizes and their rate of change. In addition the cyclic behavior of the window protocol sender causes cyclic queue activity all along the path.

Queue activity is found to have three distinct phases. The initial phase describes queue build up behavior. This phase begins with the arrival of the first bit of the initial packet at any queue. Packets arrive at a rate controlled by the previous upstream link. Queue build up continues until packet arrivals from the previ-

ous upstream resource temporarily stops. The second phase describes a short pause until arrivals begin again. Thus, any queue built up during the first phase begins draining. The third phase consists of a queue finding a cyclic pattern of packet arrivals from a previous resource. Solutions for the occurrence of each phase can be obtained through an iterative process. This process involves solving for the same information in the previous resource queues back to the base case of the window protocol sender.

Additional results show the behavior of window protocols often forces large queues to appear near a window protocol sender during initial protocol activity. At each queue, the maximum queue size occurs at or right after queue depletion of the previous upstream resource. Thus queues always drain and appear further "downstream" as data transfer continues. We refer to this activity as queue migration. The speed at which a particular queue drains is called the Queue Drain Rate. This rate is shown to be a function of the speed of the resource the queue is feeding and of the bottleneck speed. Queues can be considered migrating at the Queue Drain Rates of the various resources. Queue migration continues until the bottleneck is reached. At this point in time, if the window size is large enough, a large queue can be (and often is) permanently maintained at the bottleneck. This behavior agrees with similar behavior described by finite population closed queueing systems. These systems observe that at steady state you are most likely to find a queue in front of the bottleneck resource.

Steady state begins once sender transmission becomes cyclic at the bottleneck rate. The queue migration process begins at this same time. One intriguing result is that once the sender enters steady state, the total queue time along the path for the request packets is an invariant. This is true even while queue migration is still occurring.

It is interesting to note that despite of the wide spread use of window protocols no deterministic analysis of their queueing behavior seems to exist. Yet, the approach taken in this research appears very promising. Because deterministic dependencies are most evident when a load exists, this deterministic analysis technique also allows the accurate determination of queueing activity during significant network load, a time network designers consider most critical. The results are applicable to the window protocol mechanisms for congestion and flow control in SNA, and TCP.